FACE: Feasible and Actionable Counterfactual Explanations

Rafael Poyiadzi
University of Bristol
Bristol, United Kingdom
rp13102@bristol.ac.uk

Kacper Sokol
University of Bristol
Bristol, United Kingdom
K.Sokol@bristol.ac.uk

Raul Santos-Rodriguez
University of Bristol
Bristol, United Kingdom
enrsr@bristol.ac.uk

Tijl De Bie
University of Ghent
Ghent, Belgium
tijl.debie@ugent.be

Peter Flach
University of Bristol
Bristol, United Kingdom
Peter.Flach@bristol.ac.uk

ABSTRACT

Work in Counterfactual Explanations tends to focus on the principle of “the closest possible world” that identifies small changes leading to the desired outcome. In this paper we argue that while this approach might initially seem intuitively appealing it exhibits shortcomings not addressed in the current literature. First, a counterfactual example generated by the state-of-the-art systems is not necessarily representative of the underlying data distribution, and may therefore prescribe unachievable goals (e.g., an unsuccessful life insurance applicant with severe disability may be advised to do more sports). Secondly, the counterfactuals may not be based on a “feasible path” between the current state of the subject and the suggested one, making actionable recourse infeasible (e.g., low-skilled unsuccessful mortgage applicants may be told to double their salary, which may be hard without first increasing their skill level). These two shortcomings may render counterfactual explanations impractical and sometimes outright offensive. To address these two major flaws, first of all, we propose a new line of Counterfactual Explanations research aimed at providing actionable and feasible paths to transform a selected instance into one that meets a certain goal. Secondly, we propose **FACE**: an algorithmically sound way of uncovering these “feasible paths” based on the shortest path distances defined via density-weighted metrics. Our approach generates counterfactuals that are coherent with the underlying data distribution and supported by the “feasible paths” of change, which are achievable and can be tailored to the problem at hand.

CCS CONCEPTS

• **Computing methodologies → Machine learning**.

KEYWORDS

Explainability, Interpretability, Counterfactuals, Black-box Models

AIES '20, February 7–8, 2020, New York, NY, USA
© 2020 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-7110-0/20/02…$15.00
https://doi.org/10.1145/3375627.3375850

ACM Reference Format:
Rafael Poyiadzi, Kacper Sokol, Raul Santos-Rodriguez, Tijl De Bie, and Peter Flach. 2020. FACE: Feasible and Actionable Counterfactual Explanations. In *Proceedings of the 2020 AAAI/ACM Conference on AI, Ethics, and Society (AIES '20), February 7–8, 2020, New York, NY, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3375627.3375850

1 INTRODUCTION

In this paper we are concerned with Counterfactual and Contrastive Explanations (CE) [16] that fall under the category of *Example-Based Reasoning*. While other approaches in the field of Machine Learning Interpretability [9, 10, 14] aim at answering: “Why has my loan been declined?”, CE aim at answering: “What do I need to do for my loan to be accepted?”

Wachter et al. [17] propose three aims of explanations with respect to their audience:

(1) to inform and help the explainee understand why a particular decision was reached,
(2) to provide grounds to contest adverse decisions, and
(3) to understand what could be changed to receive a desired result in the future, based on the current decision-making model.

Counterfactual explanations achieve all three of these aims [17]. However, a naïve application of the last one – the principle of “the closest possible world” that prescribes small changes that lead to the desired outcome – may yield inadequate results. Firstly, a counterfactual generated by a state-of-the-art explainability system is not necessarily representative of the underlying data distribution and may prescribe unachievable goals. This shortcoming is illustrated in Figure 1, where points A and B – both close to the explained data point × with respect to the l_2-norm – achieve the desired prediction, however they lie in a low-density region. This last observation undermines the practical feasibility of A and B since there are no precedents of similar instances in the data. Secondly, counterfactuals provided by current approaches may not allow for a *feasible path* between the initial instance and the suggested counterfactual making actionable recourse impractical. This argument is illustrated with point D in Figure 1, which we argue is a more actionable counterfactual than C. Both these discoveries have prompted us to establish a new line of research for Counterfactual Explanations: providing *actionable* and *feasible* paths to transform

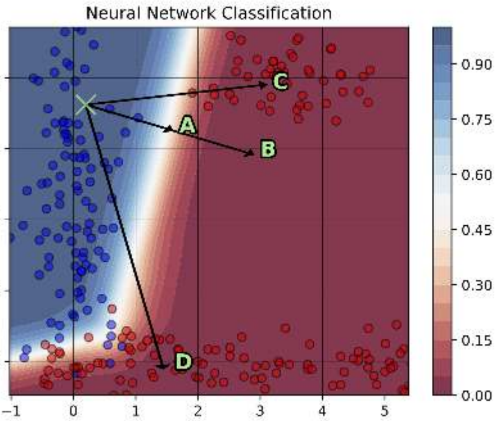


Figure 1: A**,** B**,** C **and** D **are four viable counterfactuals of** $\times$**, all satisfying the condition of having a different predicted class. We argue that** D **is the best choice.** A **is the result of minimising the** l_2**-norm and** B **is a generic data point that has a large classification margin. Nevertheless, both** A **and** B **lie in a** ***low density region.*** C **and** D **do not share the shortcomings of** A **and** B**: they lie in high-density regions and have a relatively large classification margins. The major difference between** C **and** D **is the connection between** $\times$ **and** D **via a high-density path, indicating that it is feasible for the original instance to be transformed into** D **despite** C **being simply closer.**

a certain data point into one that meets certain goals (e.g., belong to a desirable class).

The contribution of this paper is twofold. We first critique the existing line of research on Counterfactual Explanations by pointing out the shortcomings of dismissing the nature of the target counterfactual and its (real-life) context. We point out that existing research is not aligned with real-world applications (e.g., offering a *useful* counterfactual advice to customers who have been denied loans). To overcome this challenge we identify two essential properties of counterfactual explanations: *feasibility* and *actionability*, which motivate a new line of research concerned with providing high-density paths of change. Secondly, we propose a novel, well-founded approach to generating feasible and actionable counterfactual explanations that respect the underlying data distribution and are connected via high-density paths (based on the shortest path distances defined via density-weighted metrics) to the explained instance. Our approach – which we call *Feasible and Actionable Counterfactual Explanations* (**FACE**) – mitigates all of the risks associated with the explanations produced by the current line of research.

We support our claims by discussing how overlooking these premises could lead to "unachievable goals" with undesired consequences such as a loss of the end user's trust. Furthermore, we show that our algorithmic contribution to generating feasible and actionable counterfactuals is non-trivial as the resulting explanations come from dense regions and are connected with high-density paths to the original instance. Therefore, the explanations are coherent with the underlying data distribution and can be tailored to the user by customising the "feasible paths" of change. In Section 2 we establish the links and differences of our method with similar approaches in the literature. Section 3 introduces our methodology and Section 4 presents our experimental results. In Section 5 we discuss related work and in Section 6 we conclude our paper with a discussion.

2 COUNTERFACTUAL EXPLANATIONS

In this section we motivate the need for a new approach (given the current literature) that ensures usefulness of counterfactual explanations in practice. Firstly, the nature of the target instance – the derived counterfactual example – is not taken into account. This may lead to a case where the target instance is not representative of the underlying data distribution, for example, it is located in a low density region, and thus can be considered an outlier. In addition to being poor explanations, such counterfactuals are at risk of harming the explainee by suggesting a change of which the future outcome is highly uncertain, as classifiers tend to be less reliable in sparsely populated regions of the data space, especially close to a decision boundary. Points A and B shown in Figure 1 are examples of this major drawback. The uncertainty in a prediction, coming either from a low classification margin or due to low density of a region, should be of utmost importance when generating a counterfactual.

Beyond feasibility and actionability, it is also important to examine the model's confidence of predictions as it may contribute to issues with a delayed impact [8]. For example, consider a person who had his loan application rejected and wants to know what changes to make for his application to be accepted next time. If this person is handed a counterfactual explanation and implements the proposed changes, his loan application will be accepted. However, if the *new state* of the subject (the proposed counterfactual) is in a region of high uncertainty, then there exists a high risk that this individual will default.

Furthermore, the desiderata presented by Wachter et al. [17] do not account for the extent to which the change – a transformation from the current state to the suggested counterfactual state – is feasible. "Counterfactual thinking" refers to the concept of hypothesising what would have happened had something been done differently [11], i.e., "Had I done X instead of Y, would the outcome still be Z?" However, when adapting this concept to Machine Learning applications, e.g., see Robeer [11], the outcome is usually decided prior to finding a counterfactual cause. What has been overlooked by the Interpretable Machine Learning community is that the aim of a counterfactual explanation is for the explainee to *actually try and make the change* given the actionable nature of the explanation. A customer whose loan application has been rejected would (probably) disregard a counterfactual explanation conditioned on him being 10 years younger.

The current state-of-the-art solutions do not satisfy the three requirements proposed by Wachter et al. [17], which we believe are critical for actionability and thus practical utility of counterfactual explanations. To remedy this situation we propose to following objectives for counterfactual explanations in addition to the inherent requirement of these instances belonging to the desired class:

(1) feasibility of the counterfactual data point,
(2) continuity and feasibility of the path linking it with the data point being explained, and
(3) high density along this path and its relatively short length.

3 FEASIBLE COUNTERFACTUALS

Before presenting **FACE** we introduce the necessary notation and background for completeness (see Alamgir and von Luxburg [1] and references therein for an in-depth presentation of this topic). We then show how different variants of our approach affect its performance and the quality of generated counterfactuals.

3.1 Background

Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote the input space and let $\{\boldsymbol{x}_i\}_{i=1}^N \in \mathcal{X}$ be an independent and identically distributed sample from a density p. Also, let f be a positive scalar function defined on $\mathcal{X}$ and let γ denote a path connecting $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$, then the f-length of the path is denoted by the *line integral* along γ with respect to f:[1]

$$\mathcal{D}_{f,\gamma} = \int_\gamma f(\gamma(t)) \cdot |\gamma'(t)| dt. \tag{1}$$

The path with the minimum f-length is called the f-geodesic, and its f-length is denoted by $\mathcal{D}_{f,\gamma^\star}$.

Consider a geometric graph $G = (V, E, W)$ with vertices V, edges E and (edge) weights W. The vertices correspond to the sampled instances (training data) and edges connect the ones that are close with respect to a chosen metric, which value (a measure of closeness) is encoded in the (edge) weights. We use the notation $i \sim j$ to indicate a presence of an edge connecting $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, with the corresponding weight w_{ij}; and $i \nsim j$ to mark that $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are not directly connected, in which case the weight is assumed to be $w_{ij} = 0$.

Let f depend on $\boldsymbol{x}$ through the density p with $f_p(x) := \tilde{f}(p(x))$. Then, the f-length of a curve $\gamma : [\alpha, \beta] \rightarrow \mathcal{X}$ can be approximated by a Riemann sum of a partition of $[\alpha, \beta]$ in sub-intervals $[t_{i-1}, t_i]$ (with $t_0 = \alpha$ and $t_N = \beta$):

$$\hat{\mathcal{D}}_{f,\gamma} = \sum_{i=1}^{N} f_p\Big(\frac{\gamma(t_{i-1}) + \gamma(t_i)}{2}\Big) \cdot \|\gamma(t_{i-1}) - \gamma(t_i)\|.$$

As the partition becomes finer, $\hat{\mathcal{D}}_{f,\gamma}$ converges to $\mathcal{D}_{f,\gamma}$ [3, Chapter 3]. This suggests using weights of the form:

$$w_{ij} = f_p\Big(\frac{\boldsymbol{x}_i + \boldsymbol{x}_j}{2}\Big) \cdot \|\boldsymbol{x}_i - \boldsymbol{x}_j\|,$$
$$\text{when} \quad \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq \epsilon.$$

In the case that the condition does not hold, $w_{ij} = 0$. The true density p is rarely known but Sajama and Orlitsky [13] show that using a *Kernel Density Estimator* (KDE) $\hat{p}$ instead will converge to the f-distance. Sajama and Orlitsky [13] also show how to assign weights to edges while avoiding the need to perform density estimation altogether. Their results apply to two graph constructions, namely, a k-NN graph and an ϵ-graph. In summary, for the three approaches the weights can be assigned as follows:

$$w_{ij} = f_{\hat{p}}\Big(\frac{\boldsymbol{x}_i + \boldsymbol{x}_j}{2}\Big) \cdot \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \qquad (KDE) \tag{2}$$

$$w_{ij} = \tilde{f}\Big(\frac{r}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}\Big) \cdot \|\boldsymbol{x}_i - \boldsymbol{x}_j\|, \quad r = \frac{k}{N \cdot \eta_d} \qquad (k\text{-NN}) \tag{3}$$

$$w_{ij} = \tilde{f}\Big(\frac{\epsilon^d}{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|}\Big) \cdot \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \qquad (\epsilon\text{-graph}) \tag{4}$$
$$\text{when} \quad \|\boldsymbol{x}_i - \boldsymbol{x}_j\| \leq \epsilon$$

where η_d denotes the volume of a sphere with a unit radius in $\mathbb{R}^d$. Similarly to above, if the condition does not hold, $w_{ij} = 0$.

3.2 The FACE Algorithm

Building up on this background we introduce the **FACE** algorithm. It uses f-distance to quantify the trade-off between the path length and the density along this path, which can subsequently be minimised using a shortest path algorithm by approximating the f-distance by means of a finite graph over the data set. Moreover, **FACE** allows the user to impose additional feasibility and classifier confidence constraints in a natural and intuitive manner.

Firstly, a graph over the data points is constructed based on one of the three approaches: *KDE*, k-NN or ϵ-graph. The user then decides on the properties of the target instance (i.e., the counterfactual): the prediction threshold – a lower bound on prediction confidence outputted by the model, and the density (or its proxy) threshold. This part of the algorithm is described in Algorithm 1, which assumes access to a *KDE*.

To generate a counterfactual, **FACE** must be given its expected class. Optionally, the counterfactual can be additionally constrained by means of: a subjective prediction confidence threshold (t_p), a density threshold (t_d), a custom weight function (w), and a custom conditions function (c), which determines if a transition from a data point to its neighbour is feasible.[2] Subject to the new weight function and conditions function, if possible, the graph is updated by removing appropriate edges; otherwise a new graph is constructed.[3] The Shortest Path First Algorithm (Dijkstra's algorithm) [2] is executed on the resulting graph over all the candidate targets, i.e., the set $\boldsymbol{I}_{CT}$ of all the data points that meet the confidence and density requirements (see line 11 in Algorithm 1).

Complexity. Execution of the Shortest Path First Algorithm between two instances can be optimised to have the worst case time complexity of $O(|E| + |V|log|V|)$ where $|E|$ denotes the number of edges and $|V|$ the number of nodes in the graph. This complexity then scales accordingly to the number of candidate targets. The first term of the complexity – the number of edges – can be controlled by the user to a certain extent as it depends on the choice of the distance threshold parameter. The second term can also be controlled

[1] We assume that $\mathcal{X}$ is endowed with a density function p with respect to the Lebesgue measure, where p is L-Lipschitz continuous with $L > 0$.

[2] Domain knowledge of this form (e.g., immutable features such as sex or conditionally immutable changes such as age, which are only allowed to change in one direction) are incorporated within the *conditions function* $c(\cdot, \cdot)$. This knowledge is *essential* if the desired counterfactual is to be useful.

[3] If the explainee wants to provide a custom cost function for the feature value changes, e.g., the cost of changing a job is twice that of change a marital status, a new graph has to be built from scratch. If, on the other hand, the cost function stays fixed and only new constraints (inconsistent with the current graph) are introduced, e.g., the counterfactuals should not be conditioned on a job change, the existing graph can be modified by removing some of its edges.

```
Algorithm 1: FACE Counterfactual Generator
input  : Data ($\boldsymbol{X} \in \mathbb{R}^d$), density estimator ($\hat{p} : \mathcal{X} \rightarrow [0, 1]$),
         probabilistic predictor ($\boldsymbol{clf} : \mathcal{X} \rightarrow [0, 1]$), distance
         function ($d : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_{/geq0}$), distance threshold
         ($\epsilon > 0$), weight function ($w : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}_{>=0}$), and
         conditions function ($c : \mathcal{X} \times \mathcal{X} \rightarrow \{True, False\}$).
output : Graph ($V, E, W$) and candidate targets ($\boldsymbol{I}_{CT}$).

/* Construct a graph.                                              */
for every pair ($\boldsymbol{x}_i, \boldsymbol{x}_j$) in $\boldsymbol{X}$ do
    if $d(\boldsymbol{x}_i, \boldsymbol{x}_j) > \epsilon$ and $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is True then
        $i \nsim j$
        $w_{ij} = 0$
    else
        $i \sim j$
        /* In this case we use Equation 2 (KDE). This should
           be adjusted for $k$-NN and $\epsilon$-graph constructions by
           using Equation 3 and 4 respectively.                    */
        $w_{ij} = w(\hat{p}(\frac{\boldsymbol{x}_i + \boldsymbol{x}_j}{2})) \cdot d(\boldsymbol{x}_i, \boldsymbol{x}_j)$
/* Get a set of candidate targets.                                 */
$\boldsymbol{I}_{CT} = \{\}$
for $\boldsymbol{x}_i$ in $\boldsymbol{X}$ do
    if $\boldsymbol{clf}(\boldsymbol{x}_i) \geq t_p$ and $\hat{p}(\boldsymbol{x}_i) \geq t_d$ then
        $\boldsymbol{I}_{CT} = \boldsymbol{I}_{CT} \cup i$
```

(and subsequently the first term as well) by reducing the number of instances to be considered, in which case the objective would be similar to the one of "Prototype Selection". A sub-sampling as simple as a random sampling of the data points, or more sophisticated alternatives such as Maximum Mean Discrepancy [5, 6], can be used with a clear trade-off between the accuracy of the generated counterfactuals and the algorithm's speed.

In practice a base graph can be generated and stored with the most generic conditions imposed, e.g., if the data represent people, edges between people of different sex would be removed. When an explainee requests a counterfactual, he can impose further restrictions (by removing edges) to create a personalised graph, e.g., this individual is not willing to get divorced. On the other hand, if personalised cost function is required, entirely new graph needs to be generated. While the theory presented here only holds for continuous distributions, which satisfy the requirements discussed earlier, the approach can still be used with discrete features.

4 EXPERIMENTS

To empirically demonstrate the utility of **FACE** we present results of its execution on two distinct data sets. First, we show the behaviour of our algorithm on a toy data set and compare the three graph construction approaches introduced in Section 3. Secondly, we apply our algorithm to the MNIST data set [7] and show how it can be used to derive meaningful digit transformations based on the calculated path.

Synthetic Data Set. To this end, we trained a Neural Network with two hidden layers of length 10 and ReLU activation functions. **FACE** was initialised with $w(z) = -log(z)$ as the weight function and the l_2-norm as the distance function. Figures 2, 3 and 4 show the results of applying **FACE** to the toy data set when used with *KDE*, *e*-graph and *k*-NN respectively. In each, the triplet follows a similar pattern: (a) no counterfactual is generated, (b) a "good" counterfactual is generated, and (c) a "bad" counterfactual is generated. Our experimental setup adheres to a real-life use case where **FACE** is originally applied with a fairly "restrictive" configuration, which is subsequently being relaxed until a counterfactual is found. Figure 5 shows the counterfactuals found by optimising Equation 5 proposed by Wachter et al. [17], which can be compared against the ones achieved with **FACE** on the same data set (cf. Figures 2, 3 and 4).

MNIST Data Set. We applied **FACE** (based on the *k*-NN construction algorithm with $k = 50$) to two images of the zero digit from the MNIST data set [7] with the target counterfactual class set to the digit eight. The underlying predictive model is a Neural Network. Figure 6 depicts the full path from the starting instance (left) to the final counterfactual (right). The resulting path shows a smooth transformation through the zeros until an eight is reached.

5 RELATED WORK

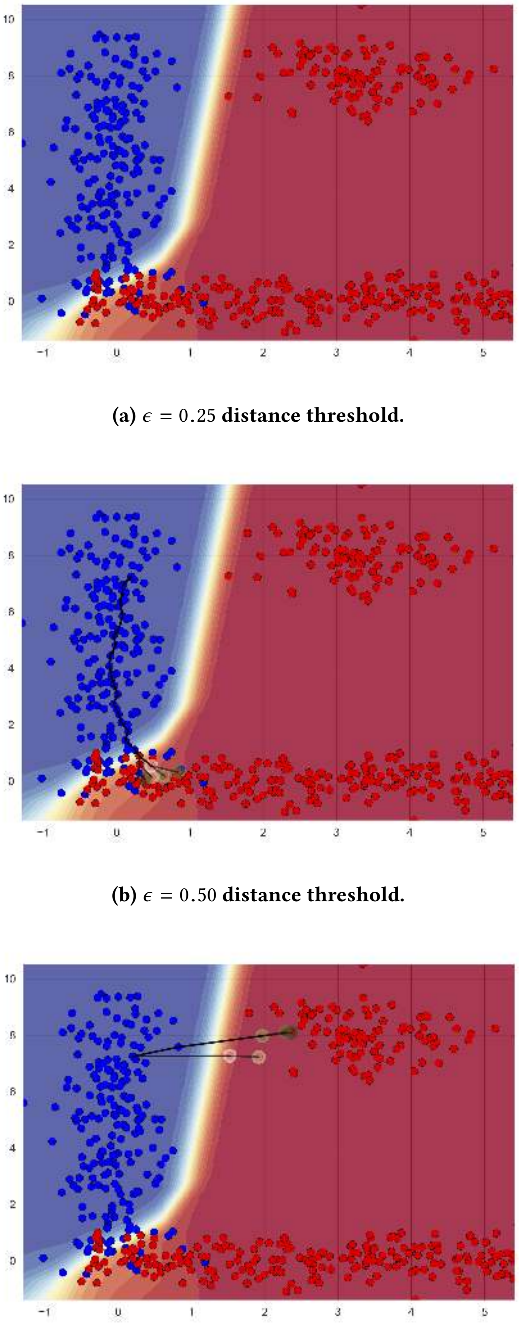
Counterfactual Explanations have been deemed to satisfy the "Right to Explanation" requirement [17] introduced by the European Union's General Data Protection Regulation (GDPR), making them viable for many businesses applying predictive modelling to human matters. To this end, Wachter et al. [17] adapted machinery used in the *Adversarial Examples* literature [4]:

$$\arg\min_{x'} \max_{\lambda} (f_w(\boldsymbol{x'}) - y')^2 + \lambda \cdot d(\boldsymbol{x}, \boldsymbol{x'}), \tag{5}$$

where $\boldsymbol{x}$ and $\boldsymbol{x'}$ denote respectively the current state of the subject and the counterfactual, y' the desired outcome, $d(\cdot, \cdot)$ a distance function and f_w a classifier parametrised by w. The objective is optimised by iteratively solving for $\boldsymbol{x'}$ and increasing λ until a sufficient solution is found. Wachter et al. emphasise the importance of the distance function choice and suggest using the l_1-norm penalty on the counterfactual, to induce sparse solutions, weighted by the *Median Absolute Deviation*. The authors deal with discrete variables by doing a separate execution of the optimisation problem, one for each unique value of every feature, and then choosing a counterfactual with the shortest distance.

Ustun et al. [15] present an Integer Programming toolkit for linear models that can be used by practitioners to analyse actionability and difficulty of recourse in a given population as well as generate advice for actionable changes (counterfactuals). Their tool "ensures recourse [actionability] in linear classification problems without interfering in model development" [15] but it does not take into account: (1) counterfactuals residing in high-density regions and (2) the existence of high-density paths connecting explained data points with counterfactual examples.

Russell [12] propose a Mixed Integer Programming (MIP) formulation to handle mixed data types and offer counterfactual explanations for linear classifiers that respect the original data structure. This formulation is guaranteed to find coherent solutions (avoiding nonsense states) by only searching within the "mixed-polytope" structure defined by a suitable choice of linear constraints. Russell

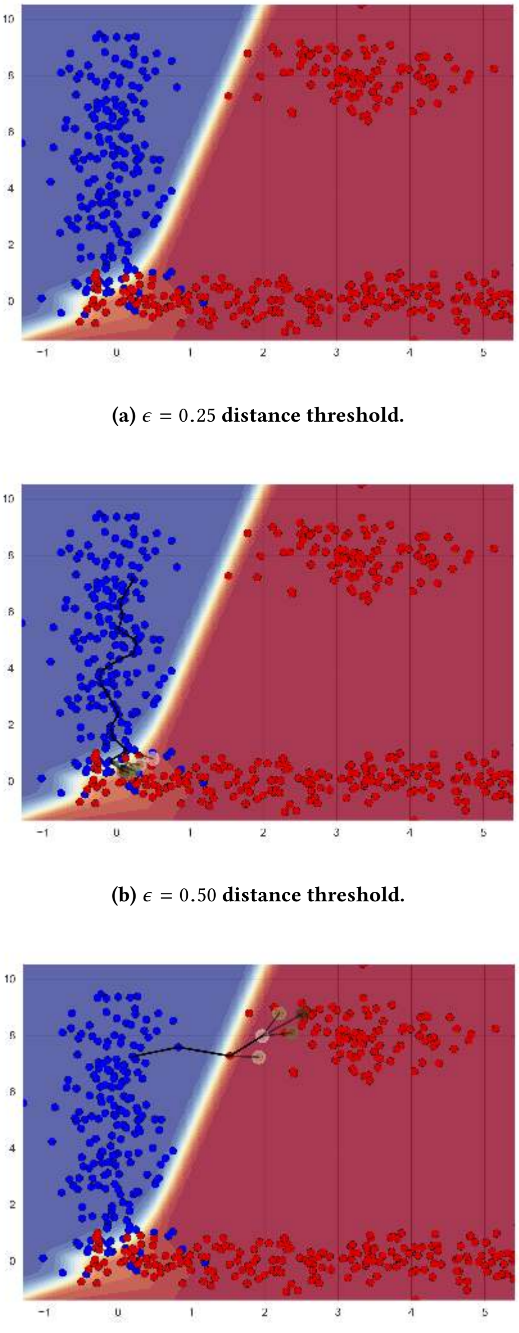


(a) $\epsilon = 0.25$ distance threshold.

(b) $\epsilon = 0.50$ distance threshold.

(c) $\epsilon = 2$ distance threshold.

Figure 2: The five shortest paths from a starting data point to a target (counterfactual) data point generated from a graph, which edge weights were computed using the KDE approach. The targets are restricted by: i) $t_p \geq 0.75$ prediction threshold, ii) $t_d \geq 0.001$ density threshold.

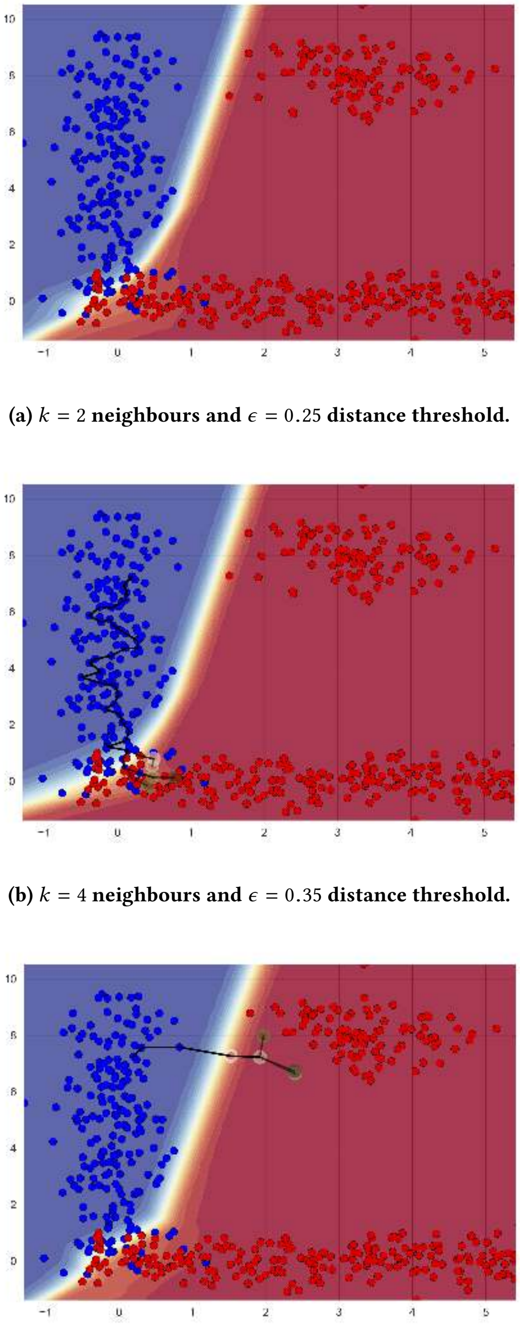


(a) $\epsilon = 0.25$ distance threshold.

(b) $\epsilon = 0.50$ distance threshold.

(c) $\epsilon = 1$ distance threshold.

Figure 3: The five shortest paths from a starting data point to a target (counterfactual) data point generated from a graph, which edge weights were computed using the e-graph approach. The targets are restricted by $t_p \geq 0.75$ prediction threshold.



(a) $k = 2$ neighbours and $\epsilon = 0.25$ distance threshold.

(b) $k = 4$ neighbours and $\epsilon = 0.35$ distance threshold.

(c) $k = 10$ neighbours and $\epsilon = 0.80$ distance threshold.

Figure 4: The five shortest paths from a starting data point to a target (counterfactual) data point generated from a graph, which edge weights were computed using the k-NN graph approach. The targets are restricted by $t_p \geq 0.75$ prediction threshold with the ϵ distance threshold and k neighbours set to: (a) $k = 2$ and $\epsilon = 0.25$; (b) $k = 4$.

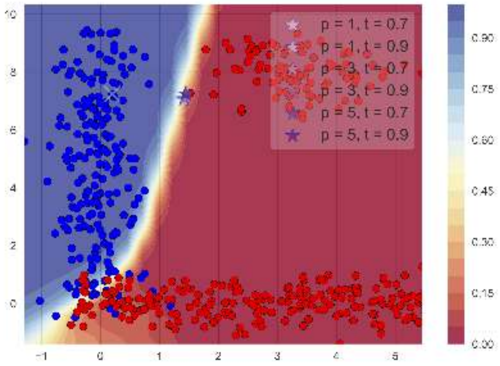


Figure 5: Counterfactuals generated using the method proposed by Wachter et al. [17]. p denotes the penalty parameter and t the classification threshold. These counterfactuals clearly do not comply with the desired properties described in Section 2.

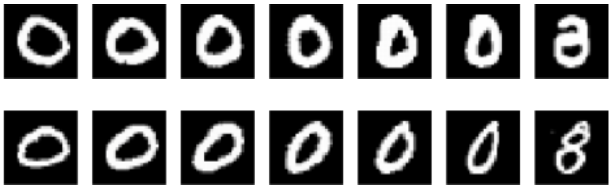


Figure 6: The FACE "transformation" paths between zero and eight (counterfactual) for two different MNIST images.

[12] chose an iterative approach to providing diverse collection of counterfactuals. Given one solution, the user can add extra constraints to the MIP that will restrict previous alterations. The list of counterfactuals is then ranked according to their l_1-distance to the explained instance.

Waa et al. [16] propose a counterfactual generation method for decision trees. Their approach uses locally trained one-vs-rest decision trees to establish a set of disjoint rules that cause the chosen instance to be classified as the target class.

FACE improves over all of the aforementioned counterfactual generation schemata in a number of ways:

- contrarily to Wachter et al. [17] and similarly to Ustun et al. [15], Russell [12] and Waa et al. [16] it supports discrete features and their restrictions in a principled manner;
- contrarily to Ustun et al. [15], Russell [12] and Waa et al. [16], and similarly to Wachter et al. [17] it is *model-agnostic*; and
- contrarily to all four approaches it produces counterfactual explanations that are both feasible and actionable.

6 SUMMARY AND FUTURE WORK

In this paper we have highlighted the shortcomings of popular Counterfactual Explanation approaches in the Machine Learning literature and proposed a new method, called **FACE**, that aims at resolving them. Our approach accounts for both the nature of the the counterfactual and the degree to which the proposed change is feasible and actionable. Our future work includes the performance evaluation of **FACE** on real-world data sets of dynamic nature and exploring the degree to which our suggested counterfactuals match the *true* change.

ACKNOWLEDGMENTS

This research was supported by the Flemish Government under the "Onderzoeksprogramma Artificiële Intelligentie (AI) Vlaanderen" programme, the EPSRC (SPHERE EP/R005273/1) and the MRC Momentum award (CUBOID MC/PC/16029).

REFERENCES

[1] M. Alamgir and U. von Luxburg. 2012. Shortest path distance in random k-nearest neighbor graphs. In *Proceedings of the 29th International Conference on Machine Learning*. International Machine Learning Society.

[2] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. 2009. *Introduction to algorithms.* MIT Press.

[3] Theodore Gamelin. 2003. *Complex analysis.* Springer Science and Business Media.

[4] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.). http://arxiv.org/abs/1412.6572

[5] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Scholkopf, and Alexander Smola. 2012. A kernel two-sample test. *Journal of Machine Learning Research* 13, Mar (2012), 723–773.

[6] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. 2016. Examples are not enough, learn to criticize! criticism for interpretability. In *Advances in Neural Information Processing Systems*. 2280–2288.

[7] Yann LeCun and Corinna Cortes. 2010. MNIST handwritten digit database. http://yann.lecun.com/exdb/mnist/. (2010). http://yann.lecun.com/exdb/mnist/

[8] Lydia T. Liu, Sarah Dean, Esther Rolf, Max Simchowitz, and Moritz Hardt. 2018. Delayed Impact of Fair Machine Learning. In *Proceedings of the 35th International Conference on Machine Learning (Proceedings of Machine Learning Research)*, Jennifer Dy and Andreas Krause (Eds.), Vol. 80. PMLR, Stockholmsmässan, Stockholm Sweden, 3150–3158. http://proceedings.mlr.press/v80/liu18c.html

[9] Scott M Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems*. 4765–4774.

[10] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. ACM, New York, NY, USA, 1135–1144. https://doi.org/10.1145/2939672.2939778

[11] Marcel Jurriaan Robeer. 2018. *Contrastive Explanation for Machine Learning.* Master's thesis. Utrecht University.

[12] Chris Russell. 2019. Efficient Search for Diverse Coherent Explanations. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT* '19)*. ACM, New York, NY, USA, 20–28. https://doi.org/10.1145/3287560.3287569

[13] Sajama and Alon Orlitsky. 2005. Estimating and computing density based distance metrics. In *Proceedings of the 22nd international conference on Machine learning*. ACM, 760–767.

[14] Kacper Sokol, Alexander Hepburn, Raul Santos-Rodriguez, and Peter Flach. 2019. bLIMEy: Surrogate Prediction Explanations Beyond LIME. *2019 Workshop on Human-Centric Machine Learning (HCML 2019) at the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada* (2019). https://arxiv.org/abs/1910.13016 arXiv preprint arXiv:1910.13016.

[15] Berk Ustun, Alexander Spangher, and Yang Liu. 2019. Actionable Recourse in Linear Classification. In *Proceedings of the Conference on Fairness, Accountability, and Transparency (FAT* '19)*. ACM, New York, NY, USA, 10–19. https://doi.org/10.1145/3287560.3287566

[16] Jasper van der Waa, Marcel Robeer, J van Diggelen, Matthieu Brinkhuis, and Mark Neerincx. 2018. Contrastive Explanations with Local Foil Trees. In *Proceedings of the ICML Workshop on Human Interpretability in Machine Learning (WHI 2018), Stockholm, Sweden, 37.*

[17] Sandra Wachter, Brent Mittelstadt, and Chris Russell. 2017. Counterfactual Explanations without Opening the Black Box: Automated Decisions and the GPDR. *Harv. JL & Tech.* 31 (2017), 841.